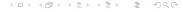
基础算法

例题·忘了来源的题目 #1

有两个长度为 $n(n \le 10^5)$ 的数列 a_1, a_2, \ldots, a_n 和 b_1, b_2, \ldots, b_n 。 每次操作可以选择一个 i(1 < i < n) 然后令 a_{i-1}, a_i, a_{i+1} 变为 $a_{i-1} + a_i, -a_i, a_{i+1} + a_i$ 。 问能否通过有限次操作将 a 变成 b。



高维前缀和

每个元素的下标为一个 k 元组 (id_1,\ldots,id_k) 和权值 w。 若干次询问,每次询问给定一个 k 元组 (q_1,\ldots,q_k) ,求所有满足 $\forall i,id_i\leq q_i$

的权值的和。 常见的情况是 $\forall i, 0 < id_i < 1$, 即子集求和。

◆□ → ◆□ → ◆ ■ → ◆ ■ ・ ◆ ● へ ● ・ ◆ ● ・ ◆ ● ・ ◆ ● ・ ◆ ● ・ ● ・ ◆ ● ● ・ ◆ ● ・ ◆ ● ● ・ ◆ ● ● ・ ◆ ● ● ・ ◆ ● ● ◆ ● ● ◆ ● ● ◆ ● ● ◆ ● ● ◆ ● ● ◆ ● ● ● ◆

例题 · CF1208F Bits And Pieces

• 给定
$$w$$
, 求 $f_S = w_S + \sum_{T \subset S} f_{T}$ 。

- 给定 w, 求 $f_S = w_S + \sum_{T \subset S} f_T$ 。
- 直接套用上面的方法肯定是不行的,不过我们可以考虑利用分治扩展。

- 给定 w, 求 $f_S = w_S + \sum_{T \subset S} f_T$ 。
- 直接套用上面的方法肯定是不行的,不过我们可以考虑利用分治扩展。
- 具体来说,我们按照二进制位分治,每一层先做左边,然后把左边的贡献到右边,再递归做右边。

- 给定 w, 求 $f_S = w_S + \sum_{T \subset S} f_T$.
- 直接套用上面的方法肯定是不行的,不过我们可以考虑利用分治扩展。
- 具体来说,我们按照二进制位分治,每一层先做左边,然后把左边的贡献到右边,再递归做右边。
- 关键在于将左边贡献到右边,而这就是朴素高维前缀和干的事情。

- 给定 w, 求 $f_S = w_S + \sum_{T \subset S} f_T$.
- 直接套用上面的方法肯定是不行的,不过我们可以考虑利用分治扩展。
- 具体来说,我们按照二进制位分治,每一层先做左边,然后把左边的贡献到右边,再递归做右边。
- 关键在于将左边贡献到右边,而这就是朴素高维前缀和干的事情。
- 复杂度相比朴素高维前缀和会增加一个 log。

例题·P1080 国王游戏

例题·忘了来源的题目 #2

有一个单调不降的函数 f 和单调不升的数组 g, 求 $\min(f(x),g(x))$ 最大值。要求做到 $O(\log V)$ 。

例题 · UOJ #750 小火车